

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claim 1. (currently amended) A method for partitioning container-managed state for a Java based application, comprising the operations of:

classifying individual entity bean objects according to ~~with~~ a particular state management type, the state management type being one of a recoverable state or a non-recoverable state, the recoverable state being one of a memory replicated state management type or a disk replicated state management type;

providing a plurality of state objects, each state object storing a state of a corresponding entity bean object within a memory address space of a Java server process, wherein each state object is associated with the state management type of the corresponding entity bean object; and

providing state management for each entity bean object using a state object corresponding to the respective entity bean object, the providing state management being based on the ~~associated~~ state management type that is associated with the state object corresponding to the respective entity bean object ~~using a corresponding state object, wherein the providing state management comprising replicating each one of the plurality of state objects is replicated in a state server, the state server for a particular state object being dedicated to a state management type that corresponds to the state management type that is associated with the particular state object~~ when dictated by the state management type.

Claims 2-3. (canceled)

Claim 4. (currently amended) A method as recited in claim 1 ~~3~~, further comprising the operation of grouping the state objects based on the type of state management to which ~~associated with~~ the corresponding entity bean object is classified.

Claim 5. (currently amended) A method as recited in claim 4, wherein the state management type into which a group of state objects are grouped identifies a policy for replication of the group of a state objects object to the dedicated state server that is dedicated to the a particular state management type corresponding to the group of state server.

Claim 6. (Original) A method as recited in claim 4, wherein the state management type identifies a policy for migration of a state object from one server process to another server process.

Claim 7. (Original) A method as recited in claim 1, further comprising the operation of managing checkpoints using the state objects.

Claim 8. (Original) A method as recited in claim 1, further comprising the operation of performing lock management using the state objects.

Claim 9. (currently amended) A method for partitioning container-managed state for a Java application, comprising the operations of:

partitioning individual entity bean objects of the Java application into state partitions, wherein the state partitions manage concurrency for the Java application, the partitioning being by state objects corresponding to the entity bean objects;

classifying individual entity bean objects within each state partition using state management units, wherein each state management unit is a collection of the state objects corresponding to one ~~identifies a~~ particular state management type for recoverable and migration capable state of the respective corresponding entity bean objects; and

~~wherein a replica of replicating each state management unit is maintained in a one of a plurality of state servers state server when dictated by the particular state management type.~~
according to the particular state management type that corresponds to the particular state objects classified in the state management unit .

Claims 10-12. (canceled)

Claim 13. (currently amended) A method as recited in claim 9 ~~12~~, further comprising the operation of using a control module to manage dynamic partitioning of the state of the application via the state partitions and the state management units.

Claim 14. (original) A method as recited in claim 13, wherein the state partitions and state management units are modular.

Claim 15. (original) A method as recited in claim 14, wherein additional state management types for the state management units can be defined.

Claim 16. (original) A method as recited in claim 15, wherein each state partition serializes transactions for entity bean objects within a particular state partition.

Claim 17. (original) A method as recited in claim 16, wherein each state partition allows only one concurrent transaction to be performed on the entity bean objects within the particular state partition during a given time period.

Claim 18. (currently amended) A system for partitioning ~~managed~~ container-managed state for a Java based application, comprising:

an application having a plurality of entity bean objects, each entity bean object comprising a state management type, the state management type being one of a recoverable state or a non-recoverable state, the recoverable state being one of a memory replicated state management type or a disk replicated state management type;

a plurality of state objects, each state object storing a state of a corresponding entity bean object within a memory address space of a Java server process, wherein each state object is associated with a particular state management type of the corresponding entity bean object; ~~and~~

a plurality of state management units that classify the state objects, a particular state object being classified into a particular state management unit based on the particular state management type of the corresponding entity bean object ~~associated with each state object,~~ wherein the state management units facilitate state management for each entity bean object; ;
~~wherein a replica of each state management unit is maintained in~~ a state server dedicated to each ~~when dictated by the particular~~ state management type, the state management type identifying a policy for replication of a state object to a state server dedicated to a particular state management type and a policy for migration of a state object from one server process to another server process; and

a replicated state manager configured to replicate a particular state management unit to the state server that is dedicated to the particular state management type of the particular state object that is classified into the particular state management unit to be replicated.

Claim 19. (original) A system as recited in claim 18, wherein the entity bean objects of the application are partitioned into state partitions during pre-deployment.

Claim 20. (original) A system as recited in claim 19, further comprising a repository that maintains state partition specifications for the state partitions.

Claim 21. (original) A system as recited in claim 20, wherein the repository manages replication and migration of state of the Java application during runtime.

Claims 22-24. (canceled)

Claim 25. (currently amended) A system as recited in claim 18, wherein ~~the state management type identifies a policy for migration of a state object from one server process to another server process.~~ the replicated state manager is further configured to replicate a particular state object from the one server process to the other server process according to the policy for migration .